



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/916,663	07/26/2001	Alain Sarraf	US20 01 0040	3478

7590 03/11/2004

Paul D. Greeley, Esq.
Ohlandt, Greeley, Ruggiero & Perle, L.L.P.
10th Floor
One Landmark Square
Stamford, CT 06901-2682

EXAMINER

MOFIZ, APU M

ART UNIT PAPER NUMBER

2175

DATE MAILED: 03/11/2004

9

Please find below and/or attached an Office communication concerning this application or proceeding.

7

Office Action Summary

Application No.

09/916,663

Applicant(s)

SARRAF ET AL.

Examiner

Apu M Mofiz

Art Unit

2175

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

- 1) ☒ Responsive to communication(s) filed on 26 July 2001.
- 2a) ☐ This action is **FINAL**. 2b) ☒ This action is non-final.
- 3) ☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

- 4) ☒ Claim(s) 1-12 is/are pending in the application.
- 4a) Of the above claim(s) _____ is/are withdrawn from consideration.
- 5) ☐ Claim(s) _____ is/are allowed.
- 6) ☒ Claim(s) 1-12 is/are rejected.
- 7) ☐ Claim(s) _____ is/are objected to.
- 8) ☐ Claim(s) _____ are subject to restriction and/or election requirement.

Apu Mofiz
Apu Mofiz
Patent Examiner
Technology Center 2100

Application Papers

- 9) ☐ The specification is objected to by the Examiner.
- 10) ☒ The drawing(s) filed on 26 July 2001 is/are: a) ☒ accepted or b) ☐ objected to by the Examiner.
Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).
Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
- 11) ☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

- 12) ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
- a) ☐ All b) ☐ Some * c) ☐ None of:
1. ☐ Certified copies of the priority documents have been received.
2. ☐ Certified copies of the priority documents have been received in Application No. _____.
3. ☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

- 1) ☒ Notice of References Cited (PTO-892)
- 2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
- 3) ☒ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08)
Paper No(s)/Mail Date 6.

- 4) ☐ Interview Summary (PTO-413)
Paper No(s)/Mail Date. _____.
- 5) ☐ Notice of Informal Patent Application (PTO-152)
- 6) ☒ Other: See Continuation Sheet.

Continuation of Attachment(s) 6). Other: certified priority document required.

DETAILED ACTION

Claim Rejections - 35 USC § 103

1. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.

2. Claims 1-12 are rejected under 35 U.S.C. 103(a) as being unpatentable over Nodoushani et al. (U.S. Patent No. 6,563,816 and Nodoushani hereinafter) in view of Haggerty et al. ("The Benefits of CORBA-Based Network Management", Communications of the ACM, October 1998, Vol 41, No. 10, pages 73-79 and Haggerty hereinafter).

As to claim 1, Nodoushani teaches a method of operating a communication network (i.e. the end-to-end system) (Fig. 28; col 38, line 65) with a network management system (i.e. "The CPA 42 further includes a home LAN manager 389, an SNMP (simple network management protocol) agent 390 and general UNIX administration utilities 388. Multiple HLHs 20 are managed from the CPA 42 through channels 362C to HL agents 380 to support all provisioning, maintenance and testing functions.") (col 39, lines 12-17), wherein said communication network (Fig. 28; col 38, line 65) comprises a number of objects (i.e. CORBA-based managed objects) (col 2, lines 45-54; col 39, lines 24-29) on a server (i.e. "According to an aspect of the invention, a network element includes a Common Object Request Broker Architecture (CORBA)-based server, CORBA-based managed objects accessible by the CORBA-based server and a CORBA-based applications programming interface (API). The CORBA-based API is coupled to an external operations support system which can manage the plural CORBA-based managed objects directly." ... "A CORBA server 386 provides the "glue" for all management applications with information regarding provisioning 382A,

inventory 394, status 384A and alarms 396.") (col 2, lines 45-54; col 39, lines 24-29), and wherein said network management system (col 39, lines 12-17) is located on the server (col 2, lines 45-54; col 39, lines 24-29) which can be accessed by a client (i.e. *"Fig. 29 shows a configuration for CORBA-based flow through provisioning with the CORBA server 386 of CPA 42 and the aforementioned OSSs 397-1, 397-2, 397-3, 397-4."* ... *"There are several advantages of the CORBA-based configuration. For example, by basing the internal management scheme on CORBA, instead of raw memory, it is very easy to add new protocols as CORBA clients, such as BelCORE TL1 protocol, if desired to inter-operate with OSSs."*) (col 40, lines 10-27), said method of operating the communication network (Fig. 28; col 38, line 65) comprises the transmission of data between the server (col 2, lines 45-54; col 39, lines 24-29) and the client (col 40, lines 10-27) under the use of the Common Object Request Broker Architecture (CORBA) (col 2, lines 45-54; col 39, lines 24-29), said method comprising using a definition for a struct (i.e.

```
    "//module MTOAMP1 {  
  
        //  
  
        ...  
  
        //structure definitions  
  
        ...  
  
        // Attributes for the T1 interface  
  
        //  
  
            struct T1Attributes {  
  
                unsigned short ifIndex;  
  
                Boolean primary;  
  
                ...  
  
            };  
  
        ...
```

```
interface TI{  
    ...  
    TIAttributes getAttributes ();  
    ...  
};  
...
```

};") (col 46; lines 15-24; col 50, lines 15-33) which only includes at least one attribute (i.e. Boolean primary) (col 46; lines 15-24; col 50, lines 15-33) but no operation (i.e. the structure by definition does not contain any operation/method as in the prior art) (col 46; lines 15-24; col 50, lines 15-33).

Nodoushani does not explicitly teach a number of objects being stored in a database on a server.

Haggerty teaches a number of objects being stored in a database on a server (i.e. *"These applications, which include Java user interfaces and HP OpenView maps, are CORBA clients communicating with the objects residing within the second level. This level can also represent client applications that are external to the system including third-party clients. The second level represents the CORBA servers and objects that model the network resources and perform management functions on the network." ... "Logging Manager receives and stores logging information and provides log query interfaces. This is a facility to store and retrieve network management logging information such as alarms. Information is stored in a database and also archived to a file system."*) (page 75, col 1, lines 1-11; page 75, col 2, lines 21-29).

It would have been obvious to a person of ordinary skill in the art at the time of Applicant's invention to modify the teachings of Nodoushani with the teachings of Haggerty to include a number of objects being stored in a database on a server with the motivation to provide a facility to store and retrieve network management logging information such as alarms (Haggerty, page 75, col 2, lines 26-27) and to make it possible to

manage large numbers of network devices in a scalable manner (Haggerty, page 73, col 1, lines 7-9).

As to claim 2, Nodoushani teaches using a definition for an interface (col 50, lines 60-65) that only includes an operation but not necessarily any attribute (i.e.

```
"Interface RegisterClient {  
    void signOn (in TrapFielder client);  
    ...  
};" (col 50, lines 60-65).
```

As to claim 3, Nodoushani teaches including a reference to said struct in said interface (i.e.

```
"//module MTOAMP1 {  
    //  
    ...  
    //structure definitions  
    ...  
    // Attributes for the T1 interface  
    //  
        struct T1Attributes {  
            unsigned short ifIndex;  
            Boolean primary;  
            ...  
};  
    ...
```

```
interface T1{  
    ...  
    T1Attributes getAttributes ();  
    ...  
};  
...
```

};") (i.e. the above code excerpts show that the struct for the "T1Attributes" is a descriptor for a corresponding interface with the name "T1". The struct only comprises the attributes of the T1, but not its operation. The operation of the T1 i.e. the getAttributes operation is contained in the definition of the "T1". The attributes are referenced by the client via the T1Attributes, which is defined in the described struct descriptor. The attributes are therefore accessible via the T1Attributes without the need for a T1. Another interface/client needs a reference to the T1Attributes to access/store the same attribute information as that which is defined in a known interface and is essentially depended on the particular need of the client application) (col 46; lines 15-24; col 50, lines 15-33).

As to claim 4, Nodoushani teaches that an instruction from the client (i.e. "Fig. 29 shows a configuration for CORBA-based flow through provisioning with the CORBA server 386 of CPA 42 and the aforementioned OSSs 397-1, 397-2, 397-3, 397-4." ... "There are several advantages of the CORBA-based configuration. For example, by basing the internal management scheme on CORBA, instead of raw memory, it is very easy to add new protocols as CORBA clients, such as BelCORE TL1 protocol, if desired to inter-operate with OSSs.") (col 40, lines 10-27) which is directed to the struct (i.e.

```
"/module MTOAMP1 {  
    //  
    ...  
    //structure definitions  
    ...  
    // Attributes for the T1 interface
```


//

```
    struct T1Attributes {  
        unsigned short ifIndex;  
        Boolean primary;  
        ...  
    };
```

...

};") (col 46; lines 15-24; col 50, lines 15-33) on the server (i.e. "According to an aspect of the invention, a network element includes a Common Object Request Broker Architecture (CORBA)-based server, CORBA-based managed objects accessible by the CORBA-based server and a CORBA-based applications programming interface (API). The CORBA-based API is coupled to an external operations support system which can manage the plural CORBA-based managed objects directly." ... "A CORBA server 386 provides the "glue" for all management applications with information regarding provisioning 382A, inventory 394, status 384A and alarms 396.") (col 2, lines 45-54; col 39, lines 24-29), results in sending data (i.e.

```
    "//module MTOAMPI {
```

...

```
    interface T1{
```

...

```
        T1Attributes getAttributes ();
```

...

```
    };
```

...

};") (col 46; lines 15-24; col 50, lines 15-33) from the server (col 2, lines 45-54; col 39,

lines 24-29) to the client (col 40, lines 10-27).

As to claim 5, Nodoushani teaches not storing the data sent to the client on the server (i.e. *"TIAttributes getAttributes ();"*). The code excerpts show that the client invokes the *getAttributes* method and the server sends the attributes to the client in a struct but does not store them in the server.) (col 46; lines 15-24; col 50, lines 15-33)).

As to claim 6, Nodoushani teaches a computer program (col 46; lines 15-24; col 50, lines 15-33) for executing a method of operating a communication network (i.e. the end-to-end system) (Fig. 28; col 38, line 65) with a network management system (i.e. *"The CPA 42 further includes a home LAN manager 389, an SNMP (simple network management protocol) agent 390 and general UNIX administration utilities 388. Multiple HLHs 20 are managed from the CPA 42 through channels 362C to HL agents 380 to support all provisioning, maintenance and testing functions."*) (col 39, lines 12-17), when run on a data processing system such as a computer (Fig. 28), wherein said communication network (Fig. 28; col 38, line 65) comprises a number of objects (i.e. CORBA-based managed objects) (col 2, lines 45-54; col 39, lines 24-29) on a server (i.e. *"According to an aspect of the invention, a network element includes a Common Object Request Broker Architecture (CORBA)-based server, CORBA-based managed objects accessible by the CORBA-based server and a CORBA-based applications programming interface (API). The CORBA-based API is coupled to an external operations support system which can manage the plural CORBA-based managed objects directly."* ... *"A CORBA server 386 provides the "glue" for all management applications with information regarding provisioning 382A, inventory 394, status 384A and alarms 396."*) (col 2, lines 45-54; col 39, lines 24-29), and wherein said network management system (col 39, lines 12-17) is located on the server (col 2, lines 45-54; col 39, lines 24-29) which can be accessed by a client (i.e. *"Fig. 29 shows a configuration for CORBA-based flow through provisioning with the CORBA server 386 of CPA 42 and the aforementioned OSSs 397-1, 397-2, 397-3, 397-4."* ... *"There are several advantages of the CORBA-based configuration. For example, by basing the internal management scheme on CORBA, instead of raw memory, it is*

Art Unit: 2175

very easy to add new protocols as CORBA clients, such as BelCORE Tl1 protocol, if desired to inter-operate with OSSs.") (col 40, lines 10-27), said method of operating the communication network (Fig. 28; col 38, line 65) comprises the transmission of data between the server (col 2, lines 45-54; col 39, lines 24-29) and the client (col 40, lines 10-27) under the use of the Common Object Request Broker Architecture (CORBA) (col 2, lines 45-54; col 39, lines 24-29), said method comprising using a definition for a struct (i.e.

```
    “//module MTOAMP1 {  
  
        //  
  
        ...  
  
        //structure definitions  
  
        ...  
  
        // Attributes for the Tl interface  
  
        //  
  
            struct TlAttributes {  
  
                unsigned short ifIndex;  
  
                Boolean primary;  
  
                ...  
  
            };  
  
        ...  
  
        interface Tl{  
  
            ...  
  
            TlAttributes getAttributes ();  
  
            ...  
  
        };  
  
        ...
```

};") (col 46; lines 15-24; col 50, lines 15-33) which only includes at least one attribute (i.e. Boolean primary) (col 46; lines 15-24; col 50, lines 15-33) but no operation (i.e. the structure by definition does not contain any operation/method as in the prior art) (col 46; lines 15-24; col 50, lines 15-33).

Nodoushani does not explicitly teach a number of objects being stored in a database on a server.

Haggerty teaches a number of objects being stored in a database on a server (i.e. *"These applications, which include Java user interfaces and HP OpenView maps, are CORBA clients communicating with the objects residing within the second level. This level can also represent client applications that are external to the system including third-party clients. The second level represents the CORBA servers and objects that model the network resources and perform management functions on the network." ... "Logging Manager receives and stores logging information and provides log query interfaces. This is a facility to store and retrieve network management logging information such as alarms. Information is stored in a database and also archived to a file system."*) (page 75, col 1, lines 1-11; page 75, col 2, lines 21-29).

It would have been obvious to a person of ordinary skill in the art at the time of Applicant's invention to modify the teachings of Nodoushani with the teachings of Haggerty to include a number of objects being stored in a database on a server with the motivation to provide a facility to store and retrieve network management logging information such as alarms (Haggerty, page 75, col 2, lines 26-27) and to make it possible to manage large numbers of network devices in a scalable manner (Haggerty, page 73, col 1, lines 7-9).

As to claim 7, Nodoushani teaches A communication network (i.e. the end-to-end system) (Fig. 28; col 38, line 65) with a network management system (i.e. *"The CPA 42 further includes*

a home LAN manager 389, an SNMP (simple network management protocol) agent 390 and general UNIX administration utilities 388. Multiple HLHs 20 are managed from the CPA 42 through channels 362C to HL agents 380 to support all provisioning, maintenance and testing functions.") (col 39, lines 12-17), wherein said communication network (Fig. 28; col 38, line 65) comprises a number of objects (i.e. CORBA-based managed objects) (col 2, lines 45-54; col 39, lines 24-29) on a server (i.e. "According to an aspect of the invention, a network element includes a Common Object Request Broker Architecture (CORBA)-based server, CORBA-based managed objects accessible by the CORBA-based server and a CORBA-based applications programming interface (API). The CORBA-based API is coupled to an external operations support system which can manage the plural CORBA-based managed objects directly." ... "A CORBA server 386 provides the "glue" for all management applications with information regarding provisioning 382A, inventory 394, status 384A and alarms 396.") (col 2, lines 45-54; col 39, lines 24-29), wherein said network management system (col 39, lines 12-17) is located on the server (col 2, lines 45-54; col 39, lines 24-29) which can be accessed by a client (i.e. "Fig. 29 shows a configuration for CORBA-based flow through provisioning with the CORBA server 386 of CPA 42 and the aforementioned OSSs 397-1, 397-2, 397-3, 397-4." ... "There are several advantages of the CORBA-based configuration. For example, by basing the internal management scheme on CORBA, instead of raw memory, it is very easy to add new protocols as CORBA clients, such as BelCORE TLI protocol, if desired to inter-operate with OSSs.") (col 40, lines 10-27), and wherein said communication network (Fig. 28; col 38, line 65) comprises the transmission of data between the server (col 2, lines 45-54; col 39, lines 24-29) and the client (col 40, lines 10-27) under the use of the Common Object Request Broker Architecture (CORBA) (col 2, lines 45-54; col 39, lines 24-29), wherein a definition for a struct (i.e.

```
    "/module MTOAMP1 {  
  
        //  
  
        ...  
  
        //structure definitions
```

```
...  
  
// Attributes for the T1 interface  
  
//  
  
    struct T1Attributes {  
  
        unsigned short ifIndex;  
  
        Boolean primary;  
  
        ...  
  
    };  
  
...  
  
    interface T1{  
  
        ...  
  
        T1Attributes getAttributes ();  
  
        ...  
  
    };  
  
...
```

};”) (col 46; lines 15-24; col 50, lines 15-33) may be used which only includes at least one attribute (i.e. Boolean primary) (col 46; lines 15-24; col 50, lines 15-33) but no operation (i.e. the structure by definition does not contain any operation/method as in the prior art) (col 46; lines 15-24; col 50, lines 15-33).

Nodoushani does not explicitly teach a number of objects being stored in a database on a server.

Haggerty teaches a number of objects being stored in a database on a server (i.e. “These applications, which include Java user interfaces and HP OpenView maps, are CORBA clients communicating with the objects residing within the second level. This level can also represent client applications that are external to the system including third-party clients. The second level represents the CORBA servers and objects that model the network resources and perform management functions on the network.” ... “ Logging

Manager receives and stores logging information and provides log query interfaces. This is a facility to store and retrieve network management logging information such as alarms. Information is stored in a database and also archived to a file system.”) (page 75, col 1, lines 1-11; page 75, col 2, lines 21-29).

It would have been obvious to a person of ordinary skill in the art at the time of Applicant's invention to modify the teachings of Nodoushani with the teachings of Haggerty to include a number of objects being stored in a database on a server with the motivation to provide a facility to store and retrieve network management logging information such as alarms (Haggerty, page 75, col 2, lines 26-27) and to make it possible to manage large numbers of network devices in a scalable manner (Haggerty, page 73, col 1, lines 7-9).

As to claim 8, Nodoushani teaches a network management system (i.e. “*The CPA 42 further includes a home LAN manager 389, an SNMP (simple network management protocol) agent 390 and general UNIX administration utilities 388. Multiple HLHs 20 are managed from the CPA 42 through channels 362C to HL agents 380 to support all provisioning, maintenance and testing functions.*”) (col 39, lines 12-17) for a communication network (i.e. the end-to-end system) (Fig. 28; col 38, line 65), wherein said communication network (Fig. 28; col 38, line 65) comprises a number of objects (i.e. CORBA-based managed objects) (col 2, lines 45-54; col 39, lines 24-29) on a server (i.e. “*According to an aspect of the invention, a network element includes a Common Object Request Broker Architecture (CORBA)-based server, CORBA-based managed objects accessible by the CORBA-based server and a CORBA-based applications programming interface (API). The CORBA-based API is coupled to an external operations support system which can manage the plural CORBA-based managed objects directly.*” ... “*A CORBA server 386 provides the “glue” for all management applications with information regarding provisioning 382A, inventory 394, status 384A and alarms 396.*”) (col 2, lines 45-54; col 39, lines 24-29), wherein said network management system (col 39,

Art Unit: 2175

lines 12-17) is located on the server (col 2, lines 45-54; col 39, lines 24-29) which can be accessed by a client (i.e. "Fig. 29 shows a configuration for CORBA-based flow through provisioning with the CORBA server 386 of CPA 42 and the aforementioned OSSs 397-1, 397-2, 397-3, 397-4." ... "There are several advantages of the CORBA-based configuration. For example, by basing the internal management scheme on CORBA, instead of raw memory, it is very easy to add new protocols as CORBA clients, such as BelCORE Tl1 protocol, if desired to inter-operate with OSSs.") (col 40, lines 10-27), and wherein said communication network (Fig. 28; col 38, line 65) comprises the transmission of data between the server (col 2, lines 45-54; col 39, lines 24-29) and the client (col 40, lines 10-27) under the use of the Common Object Request Broker Architecture (CORBA) (col 2, lines 45-54; col 39, lines 24-29), wherein a definition for a struct (i.e.

```
    “//module MTOAMP1 {  
  
        //  
  
        ...  
  
        //structure definitions  
  
        ...  
  
        // Attributes for the T1 interface  
  
        //  
  
            struct T1Attributes {  
  
                unsigned short ifIndex;  
  
                Boolean primary;  
  
                ...  
  
            };  
  
        ...  
  
        interface T1{  
  
            ...
```


TIAttributes getAttributes ();

...

};

...

};") (col 46; lines 15-24; col 50, lines 15-33) may be used which only includes at least one attribute (i.e. Boolean primary) (col 46; lines 15-24; col 50, lines 15-33) but no operation (i.e. the structure by definition does not contain any operation/method as in the prior art) (col 46; lines 15-24; col 50, lines 15-33).

Nodoushani does not explicitly teach a number of objects being stored in a database on a server.

Haggerty teaches a number of objects being stored in a database on a server (i.e. *"These applications, which include Java user interfaces and HP OpenView maps, are CORBA clients communicating with the objects residing within the second level. This level can also represent client applications that are external to the system including third-party clients. The second level represents the CORBA servers and objects that model the network resources and perform management functions on the network." ... "Logging Manager receives and stores logging information and provides log query interfaces. This is a facility to store and retrieve network management logging information such as alarms. Information is stored in a database and also archived to a file system."*) (page 75, col 1, lines 1-11; page 75, col 2, lines 21-29).

It would have been obvious to a person of ordinary skill in the art at the time of Applicant's invention to modify the teachings of Nodoushani with the teachings of Haggerty to include a number of objects being stored in a database on a server with the motivation to provide a facility to store and retrieve network management logging information such as alarms (Haggerty, page 75, col 2, lines 26-27) and to make it possible to

manage large numbers of network devices in a scalable manner (Haggerty, page 73, col 1, lines 7-9).

As to claim 9, Nodoushani teaches an interface (col 46, lines 30-37) to a network management system (i.e. *"The CPA 42 further includes a home LAN manager 389, an SNMP (simple network management protocol) agent 390 and general UNIX administration utilities 388. Multiple HLHs 20 are managed from the CPA 42 through channels 362C to HL agents 380 to support all provisioning, maintenance and testing functions."*) (col 39, lines 12-17) of a communication network (i.e. the end-to-end system) (Fig. 28; col 38, line 65), wherein said communication network (Fig. 28; col 38, line 65) comprises a number of objects (i.e. CORBA-based managed objects) (col 2, lines 45-54; col 39, lines 24-29) on a server (i.e. *"According to an aspect of the invention, a network element includes a Common Object Request Broker Architecture (CORBA)-based server, CORBA-based managed objects accessible by the CORBA-based server and a CORBA-based applications programming interface (API). The CORBA-based API is coupled to an external operations support system which can manage the plural CORBA-based managed objects directly."* ... *"A CORBA server 386 provides the "glue" for all management applications with information regarding provisioning 382A, inventory 394, status 384A and alarms 396."*) (col 2, lines 45-54; col 39, lines 24-29), wherein said network management system (col 39, lines 12-17) is located on the server (col 2, lines 45-54; col 39, lines 24-29) which can be accessed by a client (i.e. *"Fig. 29 shows a configuration for CORBA-based flow through provisioning with the CORBA server 386 of CPA 42 and the aforementioned OSSs 397-1, 397-2, 397-3, 397-4."* ... *"There are several advantages of the CORBA-based configuration. For example, by basing the internal management scheme on CORBA, instead of raw memory, it is very easy to add new protocols as CORBA clients, such as BelCORE TLI protocol, if desired to inter-operate with OSSs."*) (col 40, lines 10-27), and wherein said communication network (Fig. 28; col 38, line 65) comprises the transmission of data between the server (col 2, lines 45-54; col 39, lines 24-29) and the client (col 40, lines 10-27)

under the use of the Common Object Request Broker Architecture (CORBA) (col 2, lines 45-54; col 39, lines 24-29), wherein the interface (col 46, lines 30-37) comprises a definition for a struct (i.e.

```
    "/module MTOAMP1 {  
  
        //  
  
        ...  
  
        //structure definitions  
  
        ...  
  
        // Attributes for the T1 interface  
  
        //  
  
            struct T1Attributes {  
  
                unsigned short ifIndex;  
  
                Boolean primary;  
  
                ...  
  
            };  
  
        ...  
  
        interface T1{  
  
            ...  
  
            T1Attributes getAttributes ();  
  
            ...  
  
        };  
  
        ...  
  
    };" ) (col 46; lines 15-24; col 50, lines 15-33) which only includes at least one
```

attribute (i.e. Boolean primary) (col 46; lines 15-24; col 50, lines 15-33) but no operation (i.e. the structure by definition does not contain any operation/method as in the prior art) (col 46; lines 15-24; col 50, lines 15-33).

Nodoushani does not explicitly teach a number of objects being stored in a database on a server.

Haggerty teaches a number of objects being stored in a database on a server (i.e. *"These applications, which include Java user interfaces and HP OpenView maps, are CORBA clients communicating with the objects residing within the second level. This level can also represent client applications that are external to the system including third-party clients. The second level represents the CORBA servers and objects that model the network resources and perform management functions on the network." ... "Logging Manager receives and stores logging information and provides log query interfaces. This is a facility to store and retrieve network management logging information such as alarms. Information is stored in a database and also archived to a file system."*) (page 75, col 1, lines 1-11; page 75, col 2, lines 21-29).

It would have been obvious to a person of ordinary skill in the art at the time of Applicant's invention to modify the teachings of Nodoushani with the teachings of Haggerty to include a number of objects being stored in a database on a server with the motivation to provide a facility to store and retrieve network management logging information such as alarms (Haggerty, page 75, col 2, lines 26-27) and to make it possible to manage large numbers of network devices in a scalable manner (Haggerty, page 73, col 1, lines 7-9).

As to claim 10, Nodoushani teaches that the said computer program (col 46; lines 15-24; col 50, lines 15-33) is stored on a data carrier (i.e. a computer or any other processor) (Fig. 28).

As to claim 11, Nodoushani teaches a computer program product (col 46; lines 15-24; col 50, lines 15-33) for executing a method of operating a communication network (i.e. the end-

Art Unit: 2175

to-end system) (Fig. 28; col 38, line 65) with a network management system (i.e. *"The CPA 42 further includes a home LAN manager 389, an SNMP (simple network management protocol) agent 390 and general UNIX administration utilities 388. Multiple HLHs 20 are managed from the CPA 42 through channels 362C to HL agents 380 to support all provisioning, maintenance and testing functions."*) (col 39, lines 12-17), when run on a data processing system such as a computer (Fig. 28), wherein said communication network (Fig. 28; col 38, line 65) comprises a number of objects (i.e. CORBA-based managed objects) (col 2, lines 45-54; col 39, lines 24-29) on a server (i.e. *"According to an aspect of the invention, a network element includes a Common Object Request Broker Architecture (CORBA)-based server, CORBA-based managed objects accessible by the CORBA-based server and a CORBA-based applications programming interface (API). The CORBA-based API is coupled to an external operations support system which can manage the plural CORBA-based managed objects directly." ... "A CORBA server 386 provides the "glue" for all management applications with information regarding provisioning 382A, inventory 394, status 384A and alarms 396."*) (col 2, lines 45-54; col 39, lines 24-29), and wherein said network management system (col 39, lines 12-17) is located on the server (col 2, lines 45-54; col 39, lines 24-29) which can be accessed by a client (i.e. *"Fig. 29 shows a configuration for CORBA-based flow through provisioning with the CORBA server 386 of CPA 42 and the aforementioned OSSs 397-1, 397-2, 397-3, 397-4." ... "There are several advantages of the CORBA-based configuration. For example, by basing the internal management scheme on CORBA, instead of raw memory, it is very easy to add new protocols as CORBA clients, such as BelCORE TL1 protocol, if desired to inter-operate with OSSs."*) (col 40, lines 10-27), said method of operating the communication network (Fig. 28; col 38, line 65) comprises the transmission of data between the server (col 2, lines 45-54; col 39, lines 24-29) and the client (col 40, lines 10-27) under the use of the Common Object Request Broker Architecture (CORBA) (col 2, lines 45-54; col 39, lines 24-29), said method comprising using a definition for a struct (i.e.

```

//module MTOAMPI {

```

```
//  
...  
//structure definitions  
...  
// Attributes for the T1 interface  
//  
    struct T1Attributes {  
        unsigned short ifIndex;  
        Boolean primary;  
        ...  
    };  
...  
interface T1{  
    ...  
    T1Attributes getAttributes ();  
    ...  
};  
...  
};" ) (col 46; lines 15-24; col 50, lines 15-33) which only includes at least one
```

attribute (i.e. Boolean primary) (col 46; lines 15-24; col 50, lines 15-33) but no operation (i.e. the structure by definition does not contain any operation/method as in the prior art) (col 46; lines 15-24; col 50, lines 15-33).

Nodoushani does not explicitly teach a number of objects being stored in a database on a server.

Haggerty teaches a number of objects being stored in a database on a server (i.e. "These applications, which include Java user interfaces and HP OpenView maps, are CORBA clients communicating with the objects residing within the second level. This level can also represent client applications

that are external to the system including third-party clients. The second level represents the CORBA servers and objects that model the network resources and perform management functions on the network.” ... “ Logging Manager receives and stores logging information and provides log query interfaces. This is a facility to store and retrieve network management logging information such as alarms. Information is stored in a database and also archived to a file system.”) (page 75, col 1, lines 1-11; page 75, col 2, lines 21-29).

It would have been obvious to a person of ordinary skill in the art at the time of Applicant's invention to modify the teachings of Nodoushani with the teachings of Haggerty to include a number of objects being stored in a database on a server with the motivation to provide a facility to store and retrieve network management logging information such as alarms (Haggerty, page 75, col 2, lines 26-27) and to make it possible to manage large numbers of network devices in a scalable manner (Haggerty, page 73, col 1, lines 7-9).

As to claim 12, Nodoushani teaches that the said computer program product (col 46; lines 15-24; col 50, lines 15-33) is stored on a data carrier (i.e. a computer or any other processor) (Fig. 28).

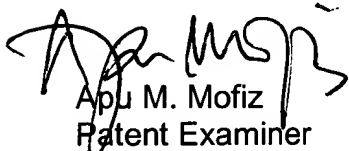
Points of Contact

3. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Apu M. Mofiz whose telephone number is (703) 605-4240. The examiner can normally be reached on Monday – Thursday 8:00 A.M. to 4:30 P.M.

Art Unit: 2175

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Dov Popovici can be reached at (703) 305-3830. The fax numbers for the group is (703) 872-9306.

Any inquiry of a general nature or relating to the status of this application should be directed to the Group receptionist whose telephone number is (703) 305-9600.



Apu M. Mofiz
Patent Examiner
Technology Center 2100

March 05,2004